Optimum Cyclic Redundancy Codes for Noisier Channels

P. Merkey
California Institute of Technology, Student

E. C. Posner
Telecommunications and Data Acquisition Office

This article considers binary cyclic redundancy codes for feedback communication over noisy digital links. The standard 16 bit ADCCP (American Data and Computer Communication Protocol) polynomial is designed for digital links which already have a low input bit error probability. For file transfer between personal computers over telephone circuits, the quality of the resulting digital circuit may be much lower. Thus we are led to consider 3 byte (24 bit) and 4 byte (32 bit) polynomials. We find generator polynomials of a certain class which have minimum weight and yet achieve the bound on minimum distance for arbitrary codes. Particular choices for 24 bit and 32 bit redundancies are exhibited: of weight and distance 6 in the 24-bit case; and weight 10 and distance 8 in the 32-bit case. This could be useful as a NASA Standard.

I. Introduction

The ADCCP 16-bit cyclic redundancy check (CRC) polynomial (Ref. 1, Sec. 4.3.1, p. 187) $p(x) = x^{16} + x^{12} + x^5 + 1$ has been specified for error detection and retransmission on digital links using the *X*.25 packet communication standard and for frame error detection in the Packet Telemetry Recommendation of the Consultative Committee on Space Data Systems. This produces a code of minimum distance 4, so it can detect three errors in a block. For this, a block may have any length (conceptually) from 17 (when one information bit is transmitted) to $2^{15} - 1 = 32{,}767$, which is one less than the period of the polynomial. Typical lengths are often in the 500 to 2500 range. Encoding of CRC polynomials is very simple algebraically, for we mrely append 16 check bits to the information bits in the unique way which makes the resulting polynomial divisible by $p(x)$ (Ref. 1, Sec. 4.1.4, p. 163). Error detections is even simpler, for, in the absence of errors, the received polynomial must be a multiple of $p(x)$.

The problem with this is that many file transfers are over voice circuits, not digital links. Thus, the input bit error probability can be much higher than for truly digital links. Furthermore, for some uses such as electronic banking, we may want extremely low output bit error probability, say 10^{-12}. For these reasons, distance-4 codes may not be powerful enough. This will mean that we must add more redundancy, for, as we show below, the largest minimum distance of an $(n, n - 16)$ code, linear or not, is 4 if $362 \leqslant n \leqslant 2^{15} - 1$.

A self-imposed constraint is to make the redundancy a multiple of 8 bits. This is because computers operate on 8-bit bytes. We may want encoding and decoding to be done in the

computers which store and use the data rather than in a special purpose outboard device. Thus we are led to look for good binary CRC codes where the redundancy, instead of being 16, is 24 or 32. This means that we want 24 and 32 degree polynomials with good distance properties. The rest of the paper finds these and shows how close they come to the largest minimum distance for any code of the same parameters, cyclic redundancy or not. Our construction will be based on known results for BCH codes or, rather, for generalized BCH codes. We recommend particular polynomials of degrees 24 and 32 which have as few nonzero terms as possible given some algebraic restrictions. This may make the programming of the encoding and decoding easier or quicker, depending on language and instruction set. Factorizations and other algebraic details of the polynomials are given as well.

II. Largest Minimum Distances

We will want to know how good our codes are, even compared with codes that don't satisfy our self-imposed constraints. By "good" we mean that our codes have close to the largest minimum distance that any code with the same redundancy can have, for a range of code lengths of interest. The following theorem allows us to find this largest minimum distance.

Theorem 1: Suppose we are given an integer $r > 0$. Let t be a non-negative integer such that $2t + 1 \leqslant r$. Let $n_b = n_b(t)$ be the smallest integer such that

$$\sum_{j=0}^{t+1} \binom{n_b}{j} > 2^r$$

Set

$$n_c(0) = \infty$$

and, for $t > 0$, let

$$n_c = n_c(t) = 2^{\lfloor (r-1)/t \rfloor} - 1$$

where $\lfloor x \rfloor$ means the greatest integer at most x. If $n_b < n_c$, then for all n such that $n_b \leqslant n \leqslant n_c$, the largest minimum distance for an $(n, n - r)$ binary code is $2t + 2$.

Remark: If $r < 2t + 1$, there is no hope of getting a minimum distance greater than or equal to $2t + 2$. Merely let exactly one of the $n - r$ information bits be 1. There are only r check bits, so if d is the minimum distance, then $d \leqslant 1 + r < 2t + 2$.

Proof: We first dispose of the case $t = 0$. Note that for $t = 0$, $n_b(0) = 2^r$. We have $n_c(0) = \infty$. The theorem holds since the Hamming Bound (Ref. 2, Chap. 1, Para. 5, p. 19) forces the minimum distance to be less than or equal to 2 for $n > 2^r$. For, if the minimum distance were 3, we would have

$$(1 + n)2^{n-r} \leqslant 2^n$$

$$1 + n \leqslant 2^r$$

$$n < 2^r$$

So the minimum distance for $n \geqslant 2^r$ is 2. We can achieve distance 2 with the code consisting of n-tuples of even parity, an $(n, n - 1)$ code. Because $r \geqslant 1$, we can also achieve distance 2 for an $(n, n - r)$ code. This completes the proof for the case $t = 0$.

Now let t be greater than 0. The Hamming Bound here too implies that the minimum distance d satisfies $d \leqslant t + 2$ if $n \geqslant n_b(t)$. For otherwise, if $d \geqslant 2t + 3$, we would have an $(n, n - r)$ $(t + 1)$-error correcting code. From this, we would find

$$\left[\sum_{j=0}^{t+1} \binom{n}{j}\right] 2^{n-r} \leqslant 2^n$$

$$\sum_{j=0}^{t+1} \binom{n}{j} \leqslant 2^r$$

But the above sum *exceeds* 2^r for $n = n_b$, and all the more for $n \geqslant n_b$. So we certainly can't do better than $d = 2t + 2$ for $m \geqslant n_b$. Can we achieve $d = 2t + 2$ if $n_b \leqslant n \leqslant n_c$? We can, as the following argument shows.

If $n \leqslant n_c(t)$ we will construct a code which has the required $d = 2t + 2$ by taking a subcode of a BCH code (Ref. 2, Chap. 7, Para. 6, p. 201). Let

$$m = \left\lfloor \frac{r - 1}{t} \right\rfloor$$

so that $m \geqslant 2$ and $n_c(t) = 2^m \geqslant 4$. Let α be a primitive element in $GF(2^m)$, and let

$$M^{(1)}(x), M^{(3)}(x), M^{(5)}(x), \ldots, M^{(2t-1)}(x)$$

be the t minimum polynomials for α, α^3, α^5, ..., α^{2t-1}, respectively. Each of these $M^{(i)}$ has degree $\leqslant m$. From Ref. 2 (Chap. 7, Para. 6, p. 201), we know that the product

$$G_{\mathrm{BCH}}(x) = M^{(1)} M^{(3)}, \ldots, M^{(2t-1)}$$

with repeated factors deleted is the generator polynomial for a minimum distance $\geqslant 2t+1$ t-error correcting BCH code. We note that deg $[G_{\mathrm{BCH}}(x)] \leqslant tm \leqslant t(r-1)/t = r-1$.

Now we define a code generator polynomial $g(x)$ of degree $r' \leqslant r$:

$$g(x) = (x+1)\, G_{\mathrm{BCH}}(x)$$

The code generated by $g(x)$ has "natural" length $2^m - 1$. Since it is a subcode of the even-weight codewords in a t-error correcting BCH code, it has minimum distance at least $2t+2$, as desired. Here $r' \leqslant r$, so we can pad in $r' - r$ check bits of 0 if we really want an $(n, n-r)$ code.

We can get a code of any other length less than 2^m by simply setting the appropriate number of high-order terms equal to zero before encoding, and then padding with $r' - r$ zeros as before. As long as we do not shorten the code to a length less than $n_b(t)$, the minimum distance $2t+2$ will be as large as possible by our earlier result. ■

We now look more closely at r, t and the set of integers $n_b(t) \leqslant n \leqslant n_c(t)$. For all we know, $n_c < n_b$ and the theorem is vacuous.

If n is such that

$$2^r < \frac{(n-t)^{t+1}}{(t+1)!}$$

then

$$\sum_{j=0}^{t+1} \binom{n}{j} \geqslant \binom{n}{t+1} \geqslant \frac{(n-t)^{t+1}}{(t+1)!} > 2^r$$

This means that $n_b(t)$ is at most such an n, or

$$n_b(t) < t + 2^{r/(t+1)}[(t+1)!)]^{1/(t+1)}$$

We note that when t is small relative to n, the usual case, this serves as a good approximation to n_b.

Now let us get a *lower* bound on $n_c(t)$. We have

$$n_c(t) = 2^{\lfloor (r-1)/t \rfloor} - 1 > 2^{[(r-1)/t]-1} - 1$$

$$= 2^{(r-t-1)/t} - 1; \quad n_c(t) > 2^{(r-t-1)/t} - 1$$

Together, the upper bound on n_b and lower bound on n_c imply the following:

$$2^{(r-t-1)/t} - 1 - t - 2^{r/(t+1)}\,((t+1)!)^{1/(t+1)} < n_c(t) - n_b(t)$$

Define $f(r, t)$ as the left-hand side of this inequality. Then

$$f(r, t) = 2^{r/(t+1)}\,(2^{[r-(t+1)^2]/[t(t+1)]} - [(t+1)!]^{1/[t+1]}) - (t+1)$$

For t fixed, then, we see that

$$f(r, t) \to \infty \quad \text{as} \quad r \to \infty$$

Specifically, since for $t \geqslant 0$

$$(t+1)! \leqslant (t+1)^{t+1}$$

we will have

$$f(r, t) \geqslant 2^{r/(t+1)}\,[2^{[r-(t+1)^2]/[t(t+1)]} - (t+1)] - (t+1)$$

So if

$$r > (t+1)^2 + (t^2+t)\log_2(t+2)$$

then $f(r, t) > 0$ and $n_b(t) < n_c(t)$. This shows that the inequality of the statement of the theorem is satisfied for a non-trivial set of r and t.

We have exactly computed the interval $n_b(t) \leqslant n \leqslant n_c(t)$ for $r = 16$, 24 and 32 in Table 1. We see from this that for codes with dimensions near 576 the minimum distances will be 4, 6, 8 for redundances 16, 24, 32, respectively. We single out $n = 576$ (72 bytes) because some file-transfer protocols for personal computers use blocks with 68 information bytes, which implies codewords of 72 bytes if $r = 32$.

III. Minimum-Weight Generator for $r = 32$

To actually find a CRC code rather than just any code presents a minor annoyance, because using the relation $r' \leqslant r$ in the preceding section, we may have to pad with zeros. We will handle this in a somewhat ad hoc fashion.

We saw from Table 1 that for $568 \leqslant n \leqslant 1023$; we can generate a distance $d = 8$ ($t = 3$) optimum code using the generator polynomial

$$g(x) = (x + 1)\,G_{\mathrm{BCH}}(x)$$

(here the degree of $G_{\mathrm{BCH}}(x)$ is 30). This is because of the following reasoning. We find that

$$m = \left\lfloor \frac{32 - 1}{3} \right\rfloor = 10$$

If α is a primitive 2^{10} th root of unity over $GF(2)$, then we seek the degrees of the minimal polynomials of α (this degree is of course 10), α^3, and α^5. Now 1023 is not prime, but it turns out that α^3 and α^5 still have all 10 distinct conjugates. Thus, the three minimal polynomials $M^{(1)}(x)$, $M^{(3)}(x)$, and $M^{(5)}(x)$ are each of degree 10, and $g(x)$ is of degree 31.

To get a generator polynomial giving a CRC of degree 32, we can multiply $g(x)$ by x or by $x + 1$ to obtain a $g_1(x)$. In the former case, we will have the rightmost position always 0. In the latter case, the period of the degree-32 generator is not 1024 but 2048. Both of these "deficiencies" are irrelevant for our application, so we shall indeed work with just such $g_1(x)$'s.

Using the primitive trinomial

$$x^{10} + x^3 + 1$$

from Peterson's tables (Ref. 3, App. C, pp. 472-492), we obtain a primitive element α of $GF(2^{10})$. Now the construction of BCH codes generalizes from that used in the previous section. Specifically, let $0 \leqslant b \leqslant 2^m - 2$ and let s be prime to 2^{m-1} (=1023 in our case), where α is a primitive element in $GF(2^m)$. Consider the $2t$ elements

$$\alpha^b, \alpha^{b+s}, \alpha^{b+2s}, \cdots, \alpha^{b+(2t-2)s}, \alpha^{b+(2t-1)s}$$

Let these elements be contained in exactly t cyclotomic cosets, say

$$C_i, \quad i = 1, 3, 5, \ldots, 2t - 1$$

Let

$$M^{(i)}(x), \quad i = 1, 3, 5, \ldots, 2t - 1$$

be the t minimal polynomials of these cosets. Then the product

$$G_{\mathrm{BCH}}(x) = M^{(1)}\,M^{(3)}\,M^{(5)}, \ldots, M^{(2t-1)}$$

generates t-error-correcting BCH code.

We are interested in $m = 10$, s prime to $1023 = 3 \cdot 11 \cdot 31$, $0 \leqslant b \leqslant 1022$, $t = 3$. There are 1023 b's to check and

$$\phi(1023) = 1023\left(1 - \frac{1}{3}\right)\left(1 - \frac{1}{11}\right)\left(1 - \frac{1}{31}\right)$$

$$= 2 \cdot 10 \cdot 30 = 600$$

values of s to check. By "check" we mean to find the $M^{(i)}$ and thus the weight (number of 1's) of $(x + 1)G_{\mathrm{BCH}}(x)$, or rather the minimum of the weights of $x(x + 1)G_{\mathrm{BCH}}(x)$ and $(x + 1)^2 G_{\mathrm{BCH}}(x)$, so that we get a polynomial of the desired degree 32.

The degree is 32, because in this case each $M^{(i)}$ has degree exactly 10. For, if one ever had degree less than 10, it would have degree at most 5, being a divisor of 10. The resulting $G_{\mathrm{BCH}}(x)$ would generate a code of length 1023, of distance $\geqslant 8$, with an $r \leqslant 26$. But calculations based on Theorem 1 imply that a code with $r \leqslant 26$ and of length 1023 cannot have minimum distance $\geqslant 8$.

Why is the code generated by $(x + 1)G_{\mathrm{BCH}}(x)$ of length 1023? If the 6 elements

$$\alpha^b, \alpha^{b+s}, \ldots, \alpha^{b+5s}$$

were all in a smaller field, then the element

$$\alpha^s = \frac{\alpha^{b+s}}{\alpha^b}$$

would be in that field as well. But s is prime to $n = 1023$, so α^s generates all of $GF(2^{10})$ and is not in a smaller subfield.

It looks as if we have $600 \cdot 1023 = 613{,}800$ polynomials (times 2 because of the x or $x + 1$ multiplier choice) of $g(x)$ to

check for weight. However, if (using the normal abuse of notation)

$$b,\ b+s,\ b+2s,\ b+3s,\ b+4s,\ b+5s$$

are in C_1, C_3, C_5, then so are

$$2b,\ 2b+2s,\ 2b+4s,\ 2b+6s,\ 2b+8s,\ 2b+10s$$

And since $(n, s) = 1$ implies $(n, 2s) = 1$ for n odd, we need only check the cases where b is a coset representative.

Further, if $(n, s) = 1$, then $(n, n - s) = 1$. So if

$$b,\ b+s,\ b+2s,\ b+3s,\ b+4s,\ b+5s$$

are in C_1, C_3, C_5, then

$$-b,\ -b-s,\ -b-2s,\ -b-3s,\ -b-4s,\ -b-5s$$

are in $\widetilde{C}_1, \widetilde{C}_3, \widetilde{C}_5$, where $\widetilde{C}_1, \widetilde{C}_3, \widetilde{C}_5$ contain the inverses of the elements in the cosets C_1, C_3, C_5. Thus the $G_{BCH}(x)$ resulting from the choices $-b$ and $-s$ will be the reciprocal of the $G_{BCH}(x)$ resulting from the choices b and s. When these reciprocals are multiplied by $(x + 1)$, the results are again reciprocals. Then, since multiplying by x doesn't change the weight and multiplying by $(x + 1)$ again preserves reciprocals, the polynomials resulting from b and s will have the same weights as those resulting from $-b$ and $-s$.

These considerations reduce our list of b's to only 55 that need to be checked. The polynomials in this list were enumerated and the lowest weight polynomials found were of weight 10. They are:

$$\left.\begin{aligned} g(x) &= x^{32}+x^{30}+x^{22}+x^{15}+x^{11}+x^7+x^6+x^5+x \\ &= x(x+1)(x^{10}+x^8+x^5+x^4+1)(x^{10}+x^7+x^6+x^3+1) \\ &\quad \times (x^{10}+x^9+x^8+x^6+x^2+x+1) \end{aligned}\right\} \text{(i)}$$

$$\left.\begin{aligned} g(x) &= x^{32}+x^{27}+x^{23}+x^{18}+x^{15}+x^{14}+x^{11}+x^8+x^4+x \\ &= x(x+1)(x^{10}+x^5+x^3+x^2+1)(x^{10}+x^9+x^5+x^4+1) \\ &\quad \times (x^{10}+x^6+x^5+x^3+x^2+x+1) \end{aligned}\right\} \text{(ii)}$$

We defined α as a root of $x^{10} + x^3 + 1$. Using this,

(1) α^{29} is a root of $x^{10}+x^8+x^5+x^4+1 = M^{(1)}(x)$

α^{87} is a root of $x^{10}+x^7+x^6+x^3+1 = M^{(3)}(x)$

α^{145} is a root of $x^{10}+x^9+x^8+x^6+x^2+x+1 = M^{(5)}(x)$

(2) α^{101} is a root of $x^{10}+x^5+x^3+x^2+1 = M^{(1)}(x)$

α^{303} is a root of $x^{10}+x^9+x^5+x^4+1 = M^{(3)}(x)$

α^{505} is a root of $x^{10}+x^6+x^5+x^3+x^2+x+1 = M^{(5)}(x)$

We recommend that one of these be adopted, specifically the second, because the gaps are more uniform. This may make hardware or software easier.

Of course, we had hoped to find a polynomial of weight 8; we do not know whether or not such a polynomial of any form exists. This seems hard to rule out.

IV. The Case r= 24

A similar search was carried out using irreducible polynomials of degree 11. We found

$$\left.\begin{aligned} g(x) &= x^{24}+x^{21}+x^{20}+x^{17}+x^{13}+x^{12}+x^3+1 \\ &= (x+1)^2(x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^2+1) \\ &\quad \times (x^{11}+x^{10}+x^9+x^7+x^6+x^5+x^4+x^3+1) \end{aligned}\right\} \text{(i)}$$

$$\left.\begin{aligned} g(x) &= x^{24}+x^{22}+x^{12}+x^{10}+x^9+x^2+x+1 \\ &= (x+1)^2(x^{11}+x^9+1)(x^{11}+x^9+x^7+x^5+x^3+x+1) \end{aligned}\right\} \text{(ii)}$$

Both have weight 8 and give the largest minimum distance 6 for lengths n between 446 and 2047. We will get a weight-6 below, but only for n up to 1023. Here, α is a primitive element of $GF(2^{11})$, a root of the polynomial

$$x^{11}+x^2+1$$

Then,

(1) α^{163} is a root of $x^{11}+x^{10}+x^9+x^8+x^7+x^6+x^5+x^2+1 = M^{(1)}(x)$

α^{489} is a root of $x^{11}+x^{10}+x^9+x^7+x^6+x^5+x^4+x^3+1 = M^{(3)}(x)$

(2) α^{341} is a root of $x^{11}+x^9+x^7+x^5+x^3+x+1 = M^{(1)}(x)$

α^{1023} is a root of $x^{11}+x^9+1 = M^{(3)}(x)$

As an alternative approach, using irreducibles of degree 10, we found

$$g(x) = x^{24} + x^{23} + x^{14} + x^{12} + x^8 + 1$$

$$= (x^3 + x^2 + 1)(x + 1)(x^{10} + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1)$$

$$\times (x^{10} + x^9 + x^6 + x^4 + 1)$$

which (with its reciprocal) was the only polynomial of this type of weight 6. Again it gives a distance 6 code but it can be used only up to a length of 1023. It does, however, have weight equal to distance, 6 instead of 8, which is satisfying. Note that this time we pad the degree to 24 with $x^3 + x^2 + 1$, which complicates the periodicity or natural length. But the argument that the distance is 6 out to length 1023 and not beyond still works. We recommend this for $r = 24$ check bits if codewords of length around 600 are desired.

As before, α is a root of $x^{10} + x^3 + 1$, which is primitive. We have

(1) α^{19} is a root of $x^{10} + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1 = M^{(1)}(x)$

(2) α^{57} is a root of $x^{10} + x^9 + x^6 + x^4 + 1 = M^{(3)}(x)$

This completes our description of the optimal polynomials for $r = 32$ and $r = 24$.

We close this article by noting that for $r = 16$, the ADCCP polynomial is the best choice. From Table 1, distance 4 is the largest attainable. The polynomial is $(x + 1)$ times a primitive, so its distance is 4. And the weight is 4. It was a good choice as a standard.

References

1. Inose, Hiroshi, 1979, *An Introduction of Digital Integrated Communications Systems*, Univ. of Tokyo Press, Tokyo, Japan.
2. MacWilliams, F. J., and Sloane, N. J. A., 1977, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam.
3. Peterson, W. Wesley, and Weldon, Jr., E. J., 1972, *Error-Correcting Codes*, Second Edition, MIT Press, Cambridge, MA.

Table 1. The interval $n_b \leq n \leq n_c$ for various r and t

Largest Minimum Distance	t	Interval of Code Lengths n		
		$r = 16$	$r = 24$	$r = 32$
$d = 4$	$t = 1$	$362 \leqslant n \leqslant 2^{15}-1$	$5793 \leqslant n \leqslant 2^{23}-1$	$92682 \leqslant n \leqslant 2^{31}-1$
$d = 6$	$t = 2$	$74 \leqslant n \leqslant 2^{7}-1$	$466 \leqslant n \leqslant 2^{10}-1$	$2954 \leqslant n \leqslant 2^{15}-1$
$d = 8$	$t = 3$	no n	no n	$568 \leqslant n \leqslant 2^{10}-1$